Large Language Models for Robot Task Allocation

Samuel A. Prieto, Member, IEEE, Borja García de Soto, Member, IEEE

***Abstract*— This study introduces an innovative framework that employs Large Language Models (LLMs) to enhance task allocation by seamlessly integrating construction robots and human users. The LLM contains key data about the task, such as agent capabilities, as well as details of the end goal to be achieved. An efficient allocation strategy is computed, balancing time efficiency and resource usage. By leveraging a Natural Language Processing interface, the system simplifies interactions with construction professionals and dynamically adjusts to unforeseen site conditions. Two LLM agents (a generator and a supervisor agent) are used concurrently to provide a more accurate task schedule. We test the proposed methodology with a simple scenario where the combination of two LLM agents provides a more accurate and logical schedule for the completion of a given task. The results highlight the significant potential of LLMs to transform operational tasks in construction, indicating a substantial step forward in aligning the industry with the latest developments in AI.**

I. INTRODUCTION

The inherent variability of construction sites, coupled with the diverse skill levels of the workforce, particularly the presence of non-skilled workers, necessitates technological solutions that are not only robust but also intuitive and adaptable. In general, the integration of robots and automated systems into daily construction tasks has been slow. The coexistence of robotic and human workforce will be achieved through a process of change [1], where new technologies will facilitate the communication and interaction between classical roles represented by construction workers (i.e., human workforce) and newly developed roles represented by robots (i.e., robotic workforce). One of the key factors to overcoming these challenges lies in the development of systems that enable flexible interaction between humans and machines, specifically through the use of natural language processing (NLP) and Large Language Models (LLM). This study considers the potential of improving efficiency and accuracy in task allocation through such technologies.

The traditional approach to task allocation and schedule optimization in construction has focused on the development of specialized tools and software designed to create the most efficient schedules based on a set task description. While these tools offer valuable resources for planning, they often lack the flexibility to adapt to the unpredicted changes that are characteristic of construction projects. In contrast, this research does not aim to replace these optimization tools but to explore a new approach and set the base for a framework capable of responding to dynamic changes with the ease and intuitiveness of natural language interaction, commanding a system composed of multiple agents of different nature (i.e., robot or human).

This paper presents a novel framework for task allocation and schedule optimization in construction settings by leveraging LLMs, such as GPT-4 [2]. The core objective of this research is to enhance the efficiency and adaptability of multi-agent systems in construction tasks. The proposed system utilizes the advanced NLP capabilities of LLMs to facilitate real-time interaction between workers and advanced technologies, such as AI and robots, allowing for the dynamic adjustment of task allocations in response to unforeseen site conditions and errors. By processing complex variables such as the capabilities of each agent, battery life management, and estimated time for sub-task completion, the LLM generates an optimized strategy that aims to minimize project duration while maximizing resource utilization.

A distinctive feature of the presented approach is the emphasis on user-friendly interaction through the NLP capabilities of LLMs. This will enable construction workers, regardless of their technical expertise, to effectively communicate and interact with the system, reducing the barrier to technology adoption within the industry. The flexibility offered by natural language interaction not only improves the integration of technology into daily operations but also empowers workers by making advanced tools accessible to a broader range of users.

LLMs have already been proven to be able to establish logical relationships between different tasks in order to achieve a given goal [3]. However, due to its mathematical base built on a regressive model, LLMs are not able to effectively reason and predict outcomes in the future based on a set of constraints. Recent developments have seen the utilization of multiple LLM agents to achieve more accurate outputs regarding this, with one agent correcting or supervising the output from another [4].

In addition, interaction with the LLM is heavily based on the quality of the prompting. Best practices suggest dividing the prompt into two clearly separated types of information: background information and API (i.e., set of actions or commands) information [5]. Moreover, the emergence of Multimodal LLMs is proving particularly useful in the field of robotics (e.g., for segmentation procedures) [6].

The use of LLMs in robotics is gaining momentum across various stages and applications of development, especially in the creation of algorithms for robot deployment [7]. Moreover, LLMs have demonstrated their capability in supervising tasks, such as detecting semantic anomalies in applications like autonomous driving [8]; however, they still face challenges in

S.A P and B.G.S. are with the S.M.A.R.T. Construction Research Group, Division of Engineering, New York University Abu Dhabi (NYUAD) (e-mail: samuel.prieto@nyu.edu, garcia.de.soto@nyu.edu).

predicting unlikely outcomes despite their accuracy with more probable ones [9].

This paper is structured as follows: Section II covers a brief state of the art on LLMs being used for robot task allocation and LLMs being used in the construction field. Section III discusses the proposed methodology using a theoretical example. Section IV presents the conclusion and future work.

II. BACKGROUND

A. *LLM robot task allocation*

The latest advances in LLMs have already seen their way into the robot task allocation field. Jin et al. [4] showcase a task planning system for robot manipulation that relies on a GPT model to generate a series of commands after being provided with the complete API and the desired task objective. A set of two LLM agents is used in their approach, with one of them generating code and the other correcting it, setting the ground for a framework where multiple agents are used to provide more accurate results.

Singh et al. [10] present a concept of situated-awareness in robot task planning, which uses the surrounding context to derive precise plans for accomplishing a broader task. This method capitalizes on the code completion of LLMs, with the prompts structured as incomplete code that the LLM then completes to outline the task plan.

Further investigation into task-oriented grasping is documented in [11], where the open-end semantic knowledge from an LLM is leveraged to guide a robot arm in grasping unknown objects.

B. *Large Language Models in construction applications*

Saka et al. [12] review study revealed new opportunities for GPT models throughout the project lifecycle. Their study revealed that current applications of GPT models in the construction industry are for information retrieval, scheduling, and logistics. Some of the identified challenges are hallucinations (i.e., incorrect or nonsensical output), lack of reliability, and trust.

Regarding safety and worker training, Hussain et al. [13] explore the use of gesture recognition to facilitate smoother and more natural interactions between humans and robots at construction sites, thereby enhancing safety. Similarly, Wang et al. [14] report the development of a chatbot designed for evaluating construction safety. Within the same scope, Uddin et al. [15] utilize LLMs for recognizing hazards and safety issues, alongside improving safety education and training for construction personnel.

You et al. [16] developed an agent called RoboGPT that uses LLMs for sequence planning for multi-step operations for construction tasks. Their results showed that a GPT model has the potential to understand the background logic of a sequential task, providing a viable solution.

Despite some of the discussed methods in both task allocation and construction applications presenting novel implementation of LLMs, none of those mentioned above take into consideration logical estimation into the future. This means that all their actions receive instant feedback, which might not be the case for large and complex scheduling scenarios. This study aims to overcome some of the GPT inherent limitations to achieve scheduling that considers logical estimation in the future.

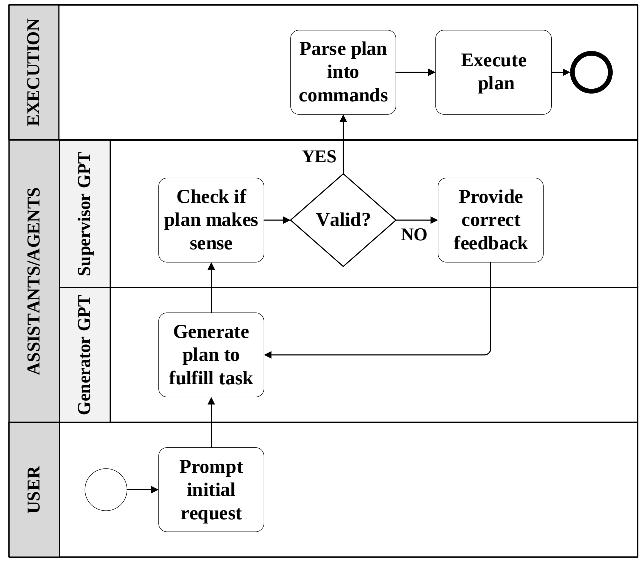

Figure 1. Flowchart of the overall interaction between the generator and the supervisor.

III. METHODOLOGY

Regressive LLMs are not capable of planning and reasoning by estimating future conditions. Their reasoning is based on the likelihood of the next word based on the previous text. To overcome this limitation, we propose a methodology based on two individual GPT agents: a Generator GPT and a Supervisor GPT.

Both agents are instructed with background information regarding the scenario, task and robots, and API information to provide clear and standardized outputs. The instructions are prompted to the GPT with a clear distinction between the background information and a set of high-level instructions and actions available from the robot, as well as a sample output of the desired result from the GPT. This ensures a more robust output from the GPT. The set of instructions is kept general without detailing specific values for the different variables. The specifics are left for the user to be prompted during the interaction with the GPT.

After being prompted with the specifics, the result from the generator, in the form of structured commands based on the API information provided on the background instructions, is fed into the supervisor for its review. The supervisor analyzes the provided schedule and, using the background information, checks if it is valid or not. In case the schedule is not valid (i.e., illogical), the supervisor also outputs a set of remarks and instructions that identify potential problems or issues with the proposed schedule and provide suggestions to make corrections. By doing that, initial issues made by the generator are spotted by the supervisor. The identified mistakes with their corresponding instructions to amend them can be fed back to the generator, resulting in a more accurate schedule.

Once the supervisor has approved the schedule, the provided API commands are parsed into instructions understandable by the robotic agents, executing the plan as intended. The output consists of a set of commands with the following fields:

{STEP #, [CURRENT_LOCATION], [ACTION], [INTERNAL_CARGO], PLACED_BRICKS, [REMAINING BATTERY]}

By having a standardized API as a result of the GPT interaction, these commands can be used as high-level instructions (i.e., move to storage) that can be matched with the low-level instructions needed for specific robotic platforms (i.e., set of velocity commands to achieve the movement). The overall methodology describing the interaction between the two GPTs is shown in Fig. 1.

IV. IMPLEMENTATION

A hypothetical scenario using GPT-4 from OpenAI is used as proof of concept of the proposed methodology. The goal is for a robot to build a brick wall consisting of four bricks. There are three distinct areas (i.e., charging area, storage area, and build area) corresponding to different actions. A schematic representation of the main elements is shown in Fig. 2. The robot needs to plan the construction of the wall by taking bricks from the storage area to the build area. Battery consumption is added as the component that requires planning since the GPT needs to account for a battery threshold needed to go back to the charging area when planning the different steps.

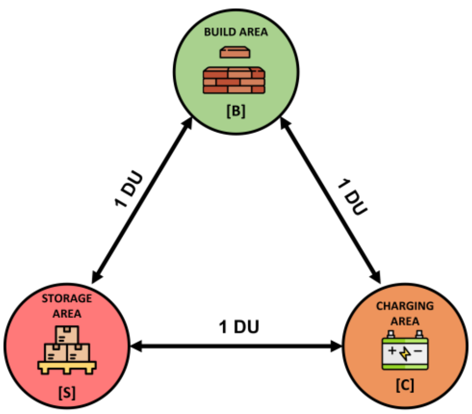

Figure 2. Layout of the developed hypothetical scenario (DU stands for distance unit).

Since the Generator GPT has a set of generic instructions defined, the user needs to prompt specific variables (i.e., actions' cost) such as the size of the wall to be built and the different costs associated with the available actions. Key variables are shown in Table 1.

TABLE I. SUMMARY OF THE ACTIONS' COST

Action	Cost
Battery consumption	20% per DU
Robot speed	1 DU/TU
Collecting 1 MU	1 TU
Installing 1 MU	1 TU
Full battery recharge	1 TU

For this theoretical scenario, generic distance, time and material units are used (DU, TU, and MU, respectively).

The output from the Generator GPT (Fig. 3) contains a total of 20 steps, which is the estimated minimum number of steps needed to complete the task based on the initial assessment of the GPT. The succession of tasks is logical and reasonable, but, as expected, the battery management fails at STEP 9, where the GPT instructs the robot to move to the storage area when the robot is already at the minimum battery threshold (i.e., 20%) and suddenly places the robot in the charging area, ignoring the battery requirements to move to that area.

```
{ STEP 1, [C], [MOVE_S], [0], 0, [80]}
{ STEP 2, [S], [PICK], [1], 0, [80]}
{ STEP 3, [S], [MOVE_B], [1], 0, [60]}
{ STEP 4, [B], [BUILD], [0], 1, [60]}
{ STEP 5, [B], [MOVE_S], [0], 1, [40]}
{ STEP 6, [S], [PICK], [1], 1, [40]}
{ STEP 7, [S], [MOVE_B], [1], 1, [20]}
{ STEP 8, [B], [BUILD], [0], 2, [20]}
{ STEP 9, [B], [MOVE_S], [0], 2, [0]}
{ STEP 10, [C], [CHARGE], [0], 2, [100]}
{ STEP 11, [C], [MOVE_S], [0], 2, [80]}
{ STEP 12, [S], [PICK], [1], 2, [80]}
{ STEP 13, [S], [MOVE_B], [1], 2, [60]}
{ STEP 14, [B], [BUILD], [0], 3, [60]}
{ STEP 15, [B], [MOVE_S], [0], 3, [40]}
{ STEP 16, [S], [PICK], [1], 3, [40]}
{ STEP 17, [S], [MOVE_B], [1], 3, [20]}
{ STEP 18, [B], [BUILD], [0], 4, [20]}
{ STEP 19, [B], [MOVE_C], [0], 4, [0]}
{ STEP 20, [C], [CHARGE], [0], 4, [100]}
```

Figure 3. Initial output from the Generator GPT, with the issues highlighted in red.

The Supervisor GPT is instructed with the same background information as the Generator GPT, for it to be aware of the end goal. After being provided with the output from the generator, the supervisor identified that the plan was not valid due to issues with battery depletion before charging in STEP 9. The supervisor also provided feedback on how to correct the mistake (i.e., sending the robot to charge before going below 20% of battery). After feeding the feedback into the Generator GPT, a new schedule was provided (Fig. 4).

```
{ STEP 1, [C], [MOVE_S], [0], 0, [80]}
{ STEP 2, [S], [PICK], [1], 0, [80]}
{ STEP 3, [S], [MOVE_B], [1], 0, [60]}
{ STEP 4, [B], [BUILD], [0], 1, [60]}
{ STEP 5, [B], [MOVE_S], [0], 1, [40]}
{ STEP 6, [S], [PICK], [1], 1, [40]}
{ STEP 7, [S], [MOVE_B], [1], 1, [20]}
{ STEP 8, [B], [BUILD], [0], 2, [20]}
{ STEP 9, [B], [MOVE_C], [0], 2, [0]}
{ STEP 10, [C], [CHARGE], [0], 2, [100]}
{ STEP 11, [C], [MOVE_S], [0], 2, [80]}
{ STEP 12, [S], [PICK], [1], 2, [80]}
{ STEP 13, [S], [MOVE_B], [1], 2, [60]}
{ STEP 14, [B], [BUILD], [0], 3, [60]}
{ STEP 15, [B], [MOVE_S], [0], 3, [40]}
{ STEP 16, [S], [PICK], [1], 3, [40]}
{ STEP 17, [S], [MOVE_B], [1], 3, [20]}
{ STEP 18, [B], [BUILD], [0], 4, [20]}
{ STEP 19, [B], [MOVE_C], [0], 4, [0]}
{ STEP 20, [C], [CHARGE], [0], 4, [100]}
```

Figure 4. Corrected output from the Generator GPT after having feedback from the Supervisor GPT. Changes are highlighted in green.

As can be seen, the newly generated schedule properly accounts for battery depletion and sends the robot back to the charging area before it runs out of battery. All the instructions provided to the GPT agents and the interaction between them can be found in [17].

One of the limitations seen from this experiment is the randomness of the GPT responses if the instructions are not concise and structured enough. The quality of the response is heavily affected by the quality of the initial prompt containing the instructions for both GPT agents.

The complexity of the performed experiment is not high enough to confidently say the system would work under all situations. However, it clearly shows that the results are better with the interaction of the two agents, with each scenario needing its own set of tailored initial instructions.

If the instructions provided by the supervisor in the first loop consistently diverge from the correct answer, the responses from the GPT agents will continue drifting further from the correct answer, highlighting a limitation inherent in the regressive behavior of how the LLM operates.

V. Conclusion and Future Work

This study demonstrates the capability of an LLM-based strategy through the utilization of multiple GPT agents to surpass the LLM limitations traditionally associated with forecasting future outcomes through logical deduction. This approach underscores the potential of LLMs to provide dynamic and adaptive solutions in complex environments where predictive accuracy is crucial. The exploration into this field reveals the inherent flexibility of LLMs, suggesting a broad spectrum of applications that extend beyond the one explored in this study.

Further research needs to be done in the field in order to explore and push the boundaries of current research regarding generative AI and LLM to fully understand the potential applications of these technologies. The next phase of this study will expand this case study and introduce additional complexity by integrating multiple robots with specialized roles, such as differentiating robots based on their efficiency in material transportation versus those optimized for construction tasks. Additional quantitative analysis will be made to assess the possible improvements achieved by the supervisor agent. This diversification will serve as a test bed to evaluate the LLM's decision-making and problem-solving capabilities in a more complex scenario.

Introducing more than one external Supervisor GPT agent will be considered to test if the addition of more agents would prevent the system from drifting into a loop of wrong answers.

Moreover, the integration with a robot simulator is planned. This step will go beyond the theoretical model, bridging the gap between hypothetical scenarios and real feedback from the simulation. This progression is aimed at validating the LLM's applicability and reliability in practical construction management settings.

Acknowledgment

This work was partially supported by the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001, and the Sand Hazards and Opportunities for Resilience, Energy, and Sustainability (SHORES) Center, funded by Tamkeen under the NYUAD Research Institute Award CG013.

References

[1] B. García de Soto, I. Agustí-Juan, S. Joss, and J. Hunhevicz, "Implications of Construction 4.0 to the workforce and organizational structures," *Int. J. Constr. Manag.*, vol. 22, no. 2, pp. 205–217, Jan. 2022, doi: 10.1080/15623599.2019.1616414.

[2] OpenAI, "GPT-4 Technical Report," arXiv. Accessed: Nov. 30, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[3] S. A. Prieto, E. T. Mengiste, and B. García de Soto, "Investigating the Use of ChatGPT for the Scheduling of Construction Projects," *Buildings*, vol. 13, no. 4, p. 857, Mar. 2023, doi: 10.3390/buildings13040857.

[4] Y. Jin *et al.*, "RobotGPT: Robot Manipulation Learning From ChatGPT," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2543–2550, Mar. 2024, doi: 10.1109/LRA.2024.3357432.

[5] "Best practices for prompt engineering with the OpenAI API | OpenAI Help Center." Accessed: Mar. 25, 2024. [Online]. Available: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api

[6] I. Hou, O. Man, S. Mettille, S. Gutierrez, K. Angelikas, and S. MacNeil, "More Robots are Coming: Large Multimodal Models (ChatGPT) can Solve Visually Diverse Images of Parsons Problems," in *Proceedings of the 26th Australasian Computing Education Conference*, in ACE '24. New York, NY, USA: Association for Computing Machinery, Jan. 2024, pp. 29–38. doi: 10.1145/3636243.3636247.

[7] N. Patton, K. Rahmani, M. Missula, J. Biswas, and I. Dillig, "Programming-by-Demonstration for Long-Horizon Robot Tasks," *Proc. ACM Program. Lang.*, vol. 8, no. POPL, p. 18:512-18:545, Jan. 2024, doi: 10.1145/3632860.

[8] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. D. Nesnas, and M. Pavone, "Semantic anomaly detection with large language models," *Auton. Robots*, vol. 47, no. 8, pp. 1035–1055, Dec. 2023, doi: 10.1007/s10514-023-10132-6.

[9] "Event Knowledge in Large Language Models: The Gap Between the Impossible and the Unlikely - Kauf - 2023 - Cognitive Science - Wiley Online Library." Accessed: Mar. 03, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/cogs.13386

[10] I. Singh *et al.*, "ProgPrompt: program generation for situated robot task planning using large language models," *Auton. Robots*, vol. 47, no. 8, pp. 999–1012, Dec. 2023, doi: 10.1007/s10514-023-10135-3.

[11] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "GraspGPT: Leveraging Semantic Knowledge From a Large Language Model for Task-Oriented Grasping," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7551–7558, Nov. 2023, doi: 10.1109/LRA.2023.3320012.

[12] A. Saka *et al.*, "GPT models in construction industry: Opportunities, limitations, and a use case validation," *Dev. Built Environ.*, vol. 17, p. 100300, Mar. 2024, doi: 10.1016/j.dibe.2023.100300.

[13] R. Hussain *et al.*, "Conversational AI-based VR system to improve construction safety training of migrant workers," *Autom. Constr.*, vol. 160, 2024, doi: 10.1016/j.autcon.2024.105315.

[14] N. Wang and R. R. A. Issa, "Transfer Learning-Based Question Generation for Building a Construction Safety Chatbot," pp. 688–694, Jan. 2024, doi: 10.1061/9780784485231.082.

[15] S. M. Uddin, A. Albert, A. Ovid, and A. Alsharef, "Leveraging ChatGPT to Aid Construction Hazard Recognition and Support Safety Education and Training," *Sustainability*, vol. 15, no. 9, p. 7121, 2023, doi: 10.3390/su15097121.

[16] H. You, Y. Ye, T. Zhou, Q. Zhu, and J. Du, "Robot-Enabled Construction Assembly with Automated Sequence Planning Based on ChatGPT: RoboGPT," *Buildings*, vol. 13, no. 7, 2023, doi: 10.3390/buildings13071772.

[17] "Large Language Models for Robot Task Allocation (additional resources)." S.M.A.R.T. Construction Research Group, Apr. 10, 2024. Accessed: Apr. 10, 2024. [Online]. Available: https://github.com/SMART-NYUAD/LLM-for-robot-task-allocation-resources-